1.00/1.001 Introduction to Computers and Engineering Problem Solving

Fall 2002

Problem Set 10

Problem 1. Implementation of Matching Service (100%)

In this problem set, you will be implementing the matchmaking service introduced in problem set 9.

For this problem set, you have two options:

- *Option 1*: You write the code for the classes you specified and described in problem set 9. You just take all the files and methods that you have come up with, and you fill in their holes with java® code. You are welcome to make major changes to your initial design if you wish. This is indeed often the case in software development. Also remember the division of labor between you and your partner in problem set 9 and try to stick to it.

- *Option 2*: You complete the files that we gave you as solution to problem set 9. In other words, you give up your design, and use ours instead. We recommend this option for those whose design was graded as being problematic. We will try to return your graded problem set 9.

We will provide you with several sets of `males.txt` and `females.txt` files with which you can test your matching service. We suggest that you copy these files and rename them when testing. You may discover that files which contain many (or maybe even not-so-many) entries may take a very long time to complete. Don’t be too alarmed when this happens; you can simply terminate the process from Forte’s “Running” tab by right-clicking on the process name, and selecting the “Terminate” menu item.

Make sure your program has the following output:

1. Output the number of females and the number of males that were read.
2. Output the list of matches (simply outputting pairs of names is sufficient)
3. Output the list of unpaired individuals, if any.
4. Output the time in seconds (with millisecond accuracy) that it took to run your algorithm (*see below*).

You can time your algorithm by using the following code fragment:

```
import java.util.Date;

/* set up your main, etc. */
...

/* start time in milliseconds */
long startTime = new Date().getTime();

... /* run your algorithm here */

/* end time in milliseconds */
long endTime = new Date().getTime();

/* total time to run in seconds */
double timeToRunAlgorithm = (endTime-startTime)/1000.0;

... /* output the results of your algorithm here */
```

Before you start coding, remember to discuss with your partner how the work should be split up.

Turnin

Turnin Requirements

- Hardcopy and electronic copy of ALL source code (all .java files). Remember to comment your code, points will be taken off for insufficient comments.
- Place a comment with your name, username, section, TA's name, assignment number, and list of people with whom you have discussed the problem set on ALL files you submit.
- Do NOT turn in electronic or hardcopies of compiled byte code (.class files).

Collaboration

For this problem set, you may work together with your assigned partner in the class. If you choose to work with your partner, you must include both of your names on your submission. (If you have different TAs, be sure you write both TAs' names on your PS.) You will not be allowed to add the name of your partner after submitting your problem set. Only submit the problem set ONCE (choose either person). Both you and your partner will get the same grade. Only 1 or 2 member teams, no exceptions.

Electronic Turnin

Use *SecureFX* (or another secure ftp or secure shell program) to upload your problem set to your 1.00 homework locker.

Detailed instructions of how to upload are on the course website.

Since your problem set is due at the beginning of lecture, your uploaded problem should have a timestamp of no later than morning on the due date.

Penalties

- Missing Hardcopy: -10% off problem score if missing hardcopy.
- Missing Electronic Copy: -30% off problem score if missing electronic copy.
- Late Turnin: -30% off problem score if 1 day late. More than 1 day late = NO CREDIT.

If your problem set is late, or if a professor has granted you an extension in advance, do not submit a printed copy of your problem set.